基于树型结构网格矢量量化的点云渲染算法 *

石祖旭 [1]，曾　安 [1†]，Vincent Ricordel[2]，Nicolas Normand[2]

(1. 广东工业大学 计算机学院 计算机系，广州 511400; 2. 南特大学 LS2N 实验室，法国 南特 44306)

摘　要：针对 3D 模型海量点云数据存在的空间冗余问题，提出一种基于 TSLVQ (tree structure lattice vector quantization) 的静态点云有损渲染算法。算法旨在利用层级嵌套网格的集合，解决渲染低效的问题。首先对整个点云进行包围盒封装，多层量化，把数个较小尺度的截断包围盒嵌入到一个较高尺度的截断包围盒单元中，每一步量化过程采用八叉树方法将包围盒分割为 8 个最佳尺寸的空或非空小包围盒，最后在最高深度的层级里，用包围盒来代替整个小包围盒中全部的点。同时，算法可自行设定八叉树的深度，从而任意控制编码的复杂度和精度，满足渲染的实时性要求。实验结果表明，与现有的网格有损压缩算法相比，提出的算法能在保证模型重建精度的基础上，具有较好的空间分解优势，实现实时渲染效果。

关键词：向量量化；树型结构网格编码矢量量化；有损压缩；3D 点云

中图分类号：TP391.41　　doi: 10.3969/j.issn.1001-3695.2018.01.0114

Efficient representation of static point cloud by using TSLVQ

Shi Zuxu[1], Zeng An[1], Vincent Ricordel[2], Nicolas Normand[2]

(1. *College of Computer Guangdong University of Technology*, *Guangzhou* 511400, *China* 2. *LS2N Lab*, *University de Nantes*, *Nantes* 44306, *France*)

Abstract: Concerning the problems that in point cloud with a lot of temporal redundancy, this paper presented a novel lossy compression approach based on tree-Structure Lattice Vector Quantization for static point cloud. The proposed approach utilizes the hierarchical packing of embed truncated lattices, which can solve the problem of representation inefficiency. Firstly, enclose the point cloud with a cube, then the point cloud is truncated, embed smaller-scale cubes into the bigger-scale cube, the multi-stage procedure of quantization, in every level, this method divided the cube to 8 sub-cutes or voxels as empty or non-empty subspace. Finally in the predefined max level, use the cube to represent all the other points in the same cube. Furthermore, it allows for predefining max level to control coding complexity and coding precision to meet the real-time rendering requirement. Experimental results show that the octree decomposition has obvious advantages in 3D object real-time representation comparing with the existing gridding lossy compression, guaranteeing 3D model reconstruction precision.

Key words: Vector Quantization; TSLVQ; lossy compression; 3D point cloud

0　引言

近年来，三维激光扫描技术，在 3D 视频、虚拟旅游、数字城市和多媒体领域有很广泛的应用。传感器，如微软的 Kinect，立体摄像头系统或高速高分辨率激光等，通过扫描能生成大量高帧率数据。然而，产生的海量点云中包含较多的冗余数据，给存储、传输及后续的数据处理带来了极大不便，尤其在以图片内容为主要特征的互联网上更为突出，因此提出高效压缩方案变得很有必要。

对于高维度的数据集，目前的研究主要集中在基于概率的压缩[1~9]、基于曲率的压缩[10]和基于网格的压缩[11~15]。Kalaiah 和 Varshney[6]提出了基于聚类层级主成分分析（principle component analysis, PCA）的统计集合渲染方法。Hornung 等人[8]设计了一种概率八叉树的表示和相关剪枝策略，解决了存储消耗和置信度阈值问题。赵尔平等人[9] 提出邻点差值渐进压缩和基于裁剪重叠区域进行冗余处理的 R 树空间索引方法。裴书玉等人[10]提出一种基于自适应分层的文物点云压缩算法。Weir 等人[11]将点云划分为若干个小包围盒，用每个小包围盒中最靠

收稿日期：2018-01-09；**修回日期：**2018-03-21　**基金项目：**国家自然科学基金资助项目（61772143，61300107，61672168）；广州市科技计划资助项目（201601010034，201804010278）；广东省大数据分析与处理重点实验室开放基金资助项目（201801）

作者简介：石祖旭（1988-），男，湖北荆州人，硕士，主要研究方向为图像处理与模式识别；曾安，女（通信作者），教授，博士，主要研究方向为人工智能，数据挖掘和概率统计理论与方法的研究及其应用（437871335@qq.com）；Vincent Ricordel，男，副教授，博士，主要研究方向为视频编码和图像序列分析；Nicolas Normand，男，副教授，博士，主要研究方向为离散距离和数学形态学．

近中心的点来代替整个小包围盒中全部的点。Martin 等人[12]采用均匀网格法的思想，把点云投影到相应的网格平面中，从投影到同一个网格的点里选择一个中值点来代替这个网格，并删除其余点，实现点云精简。但如果点分布不均衡，该方法容易产生大量空栅格。孟娜[13] 采用非均匀网格法，利用八叉树原理[14] 对网格进行细分，在点数量较多的区域空间划分较为细致，反之则较粗糙。Chen 等人[15] 提出一种通过减少三角形从而达到减少数据点的方法，其基本思想是先对点云进行三角化，然后对临近三角面片的法矢量进行比较。对较为平坦的区域，在矢量加权算法的基础上使用较少的大三角面片代替较多小三角面片，实现点云精简。由于三角网格法需要对点云进行三角化处理，对于大规模散乱点云而言效率较低，因此该方法在实际应用中受到一定限制。

同时，基于光谱理论的压缩[16, 17]也成为近来一个流行的研究方向。Taubin 等人[16]将信号处理的知识引入三维模型，其主要步骤是用快速傅里叶变换将空域转为到频域上。由任意拓扑结构的网络顶点邻接矩阵及其顶点，得到网格 Laplacian 矩阵的定义及由特征向量构成的 R^n 空间的正交基底，特征向量相对应的特征值即为频率，网格表面较为光顺的区域即为低频信号，噪声对应到频域上是高频信号。该算法运用λ|μ方法，通过增加一个负收缩因子将拉普拉斯平滑引起的收缩再放大回去。王向阳等人[17] 提出了一种基于小波变换的有损图像压缩算法。该算法首先利用小波变换对图像进行分解，然后根据各子图像特点对小波系数进行相应的向量量化编码。Ochotta 和 Saupe[18]提出 3D 模型分块和表面碎片变换的概念，从而将 2D 小波技术应用于 3D 数据压缩。

鉴于当前算法的执行效率问题，本文结合 TSLVQ 提出一种适应不同分辨率，复杂度和精确度需求的可通用的，可拓展的，有损的点云流压缩方法。

1 算法原理

1.1 3D 点云的八叉树表示

一棵八叉树是一个有预定义深度的树型结构，每个分支节点代表一个在 3D 空间中的确定的立方体，也叫体素。初始情况下，采用某种分层方式将三维空间划分为若干个体素，例如，采用深度优先遍历方式构造一棵八叉树，3D 点云所有点被封装在根节点中，形成包围盒，再循环执行该过程，直到达到设定的深度为止，子节点中可以包含/不包含点云中的点。3D 点云中的所有节点最终被映射到叶节点上。图 1 展示了不同深度等级下的八叉树分解过程，其中属于同一体素的点被涂以相同颜色。每个体素包含一个或若干个点。

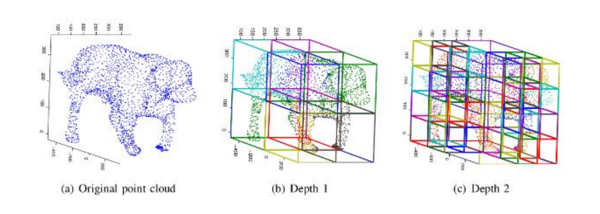


图 1 3D 模型在两个不同深度等级上八叉树分解[19]

1.2 矢量量化

矢量量化 VQ（vector quantization）是信号处理中典型的量化技术，矢量是由若干个标量数据组成的，矢量量化把矢量空间分成若干个小区域。每个小区域寻找一个代表矢量，量化时落入小区域的矢量就由这个代表矢量代替。实际上对每一个量化矢量（码字），利用其在码书中分配的索引号进行存储或传输，从而达到数据压缩的目的。图像的相邻数据项都是相关的，故而可以将其应用于图像压缩。图像编码如图所示。

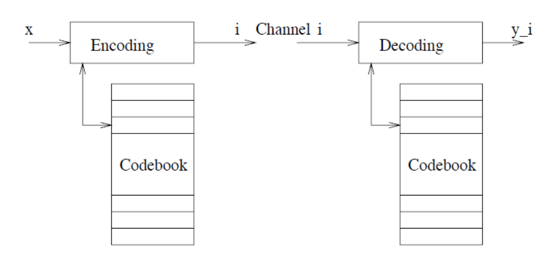


图 2.基于 VQ 的编码

输入和码本匹配的编码过程需要大量计算，解码只需查码书。一个维度为 k，容量为 L 的 VQ 函数定义为

$$Q: R^k \to D,$$
$$x \to Q(x) = y_i$$
$$D = \{y_i \in R^k / i = 1,2,\cdots,L\} \quad (1)$$

其中：x是输入向量，Q是量化函数，D是码书。编码采用“最近邻原则”，其计算公式为

$$C_i = \{x \in R^k / Q(x) = y_i, if\ d(x,y_i) \le d(x,y_j), \forall j \ne i\} \quad (2)$$

其中：C_i是Voronoï区域，$d(x,y_i)$表示输入向量和代表矢量的欧氏距离。

1.3 结构化矢量量化

结构化矢量量化（*lattice vector quantization*，LVQ）将 lattice 点集的子集作为 VQ 的输出点，通过网格编码将纯矢量量化扩展成具有更大码书尺寸的矢量量化器，因此相比 VQ 具有更好的性能。LVQ 不仅利用了矢量量化器，因此相比 VQ 具有更好的性能。LVQ 不仅利用了量化信号间的空间相关性，而且它也利用了其时间相关性，同时根据受约束的率失真理论和不受约束率失真理论，该方法原理上可以任意逼近率失真下限。码字是Voronoï区域（包围盒）的中心点。包围盒的空间嵌入决定了它的属性。不像 LBG-based (Linde, Buzo, Gray 算法) 方法[22~25]，LVQ 中的编码与矢量坐标相关。因此，它仅仅依赖于近似和尺度缩放操作，编码非常方便[23, 24]。具体步骤如下：

a）根据不同的维度，选择最合适的包围盒。与包围盒相关联的快速量化算法[26]主要采用Z^k，D_k，E_8，Λ_{16}[27]等筛选标准。

b）码书尺度的选择。

c）源数据标准化。在源数据集被标准化后，就可以使用

快速量化算法[26]。标准化操作是指将输入向量投影到截断的网格内。量化前需要进行一系列缩放操作。

d）检索包围盒码书。由于码书的尺寸非常大，对码字进行索引是复杂度很高的计算过程。标签化的目的是为了在计算复杂度和转换码书容量上折中。

文献[24]指出并不是所有的包围盒都有最佳空间嵌入。此外，量化的效率和量化区域大小和被选中的多面体形状有关，该多面体表示Voronoï区域。理论上，体积相同时，球形区域的最大粒状误差最小，球形量化区域的平均粒状误差最小。被选中的最佳包围盒具有独特的空间特性。

在三维空间中有三类包围盒：正方体包围盒（$\boldsymbol{Z_3}$），菱形十二面体（D_3）和截断八面体（$D3^*$）。图 3 表示了一个由六边形嵌入的 2D 包围盒。

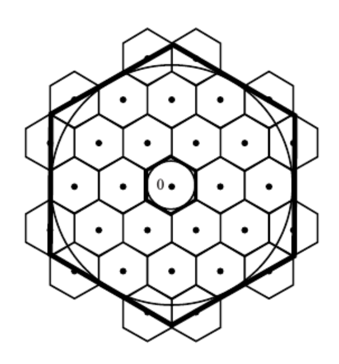
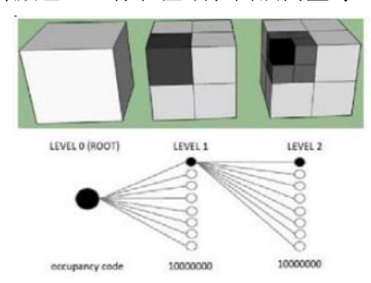


图 3　子最佳嵌入-六边形包围盒[27]

逆量化过程包括解码接收到的索引，找到对应的包围盒中心点，并再次标准化。

1.4　树结构矢量量化

树结构矢量量化（tree structure vector quantization, TSVQ）是一个多种量化方法的集合，其中的量化过程通过决策树量化[22]。该方法的优点在于运用简化版的子码书和自适应的结构简化计算。点云数据模型是一系列点，点的属性包含颜色、坐标和区域表面法向量等，如图 4 所示。



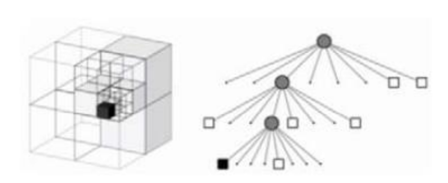


图 4　在点云中的点集合[8]

2　树型结构网格矢量量化

利用嵌入式截取网格的分层包围结构，本文提出了基于树型结构的网格矢量量化（tree structure lattice vector quantization, TSLVQ）[28]的点云渲染算法。该算法采用八叉树分割方法，在连续的层级包围盒之间设置缩放因子。

基于树型结构网格矢量量化的点云渲染算法步骤如下[27]：

a）源向量被投放到一个截断包围盒；

b）为了得到一个更好的量化，另一个较低尺度的截断包围盒被嵌入到正方体区域（*Voronoï* 区域）中，该区域是统一尺度的输入向量；

c）以上步骤循环执行。

处理统一尺度的输入向量比处理不同尺度的多个包围盒要简单很多。对应本文算法的编码流程如图 5 所示

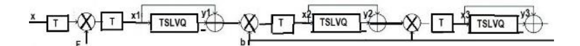


图 5　tTSLVQ 的原理

其中x是输入向量，第一个T代表平移三维空间原点到 3D 物体的中心，y_i是第 i 层的输出向量，F是初始缩放因子，b表示后续缩放因子。$b = 2$的层级如图 6 所示。

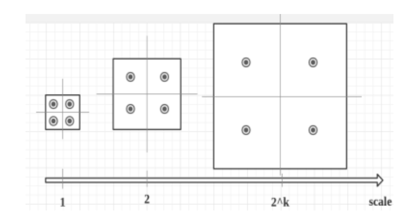


图 6　层级集合对应于包围盒

图 5 第二个 T 代表 2x2x2 到 3x3x3 坐标变换过程，如图 7 所示

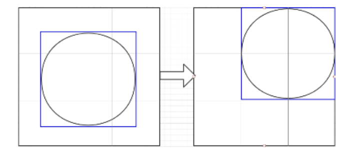


图 7. 坐标平移变换

图 7 展示了缩放后的点的三维坐标与 center=[0.5,0.5,0.5]求和之后的平移过程，图 5 中的 F 用于映射每一个输入向量x到第一个截断包围盒，其计算公式为

$$F = \frac{b \times \rho}{L_{2max}} \tag{3}$$

其中：ρ是对应的填充半径，L_{2max}为x的最大的球体半径。所有的输入向量被投射到一个超球体中，这个球体的半径等于$b \times \rho$。

在这个标准化的空间里，缩放因子 b 用于映射每个转换的向量到另一个层级截断包围盒。在渲染算法中，i^{th}层这个截断包围盒的输出向量是 y_i。与 x 有关的输出向量的最终结果计算公式为

$$y = \frac{1}{F} \times \sum_{i} \frac{y_i - Center}{b^{i-1}} \tag{4}$$

在每一层，相同的快速量化算法被运用。基于树型结构网格矢量量化的点云渲染算法伪代码如下：

```
算法 1 基于树型结构网格矢量量化的点云渲染算法
输入 3D点云模型D
输出 压缩渲染点云G
function PROCESS(D, level, final)
    initial parameters
    D ← D − mean(D)
    L_2max radius of the sphere of root
    F ← (b×ρ)/L_2max
    for all x such that x ∈ D_l do
        D_l = x * F
    end for
    while level < final do
        D_l+1 = RENDER(D_l, level, b, final)
        D_l = D_l+1
        level ← level + 1
    end while
end function
```

```
算法 2 点云分解中间过程
输入 3D点云当前层D_l
输出 3D点云下一层D_l+1
function RENDER(D_l, level, b, final)
    Center ← [0.5, 0.5, 0.5]
    for all x such that x ∈ D_l do
        D_l = x * b
        D_l = D_l + Center
        Y_l = Quantization(D_l)
        index_l = codeBook(Y_l)
        Y_l = reverseCodeBook(index_l)
        y = 1/F × Σ_l (Y_l − Center)/b^(l−1)
        plot cube G_l in the center of y
        D_l+1 = ((D_l − Y_l) × b)
    end for
    return D_l+1
end function
```

算法 2 中 *Quantization(D_I)* 为量化函数，*codeBook(Y_I)* 码字在码书中的索引查询函数，*reverseCodeBook($index_i$)* 索引在码书中的码字查询函数。

3　实验结果与分析

为了客观地评估本文算法的有效性，在两个标准的公开点云库（MIT CSAIL 图像库[30]的石狮和斯坦福大学图像学实验室[31]的中国龙）、3D 模型太阳花和 3D 潜艇（通过三维重建软件 photoscan 从 100 多张二维图片上分别构建的图像库）上进行实验验证。实验运行的硬件环境为 Intel Core i5/双核四线程 2.50GHz CPU，内存为 8GB，软件平台为 MatlabR2017a。

在点云数据精简评价中大多数采用目视的方式，也有部分通过计算 Rate-Distortion 曲线来评价。本文采用体积变化比率和渲染时间开销来评价算法的性能。在试验中，本文算法采用基于八叉树结构的空间分割，设置缩放因子 b=2。同时，与文献[29]分割点云所在空间 4 次后的结果。

图 9~12 分别展示了本文算法和文献[29]在石狮、中国龙、太阳花和潜艇数据集上处理的第一层、第二层和第三层的点云所在空间分割结果。

假设在本文算法和文献[28]提出的实验方案中，每一次都完全分割，则分别构成了完全八叉树和完全 27 叉树。表 1 中显示了两棵完全树在不同层级上的包围盒数量。

表 1　完全 27 叉树和完全八叉树对比分析

深度/节点数	八叉树	27 叉树
0	1	1
1	8	27
2	64	729
3	512	19683
4	4096	531441
5	32768	143448907
6	262144	
7	2097152	
8	16777216	

就真实渲染效果而言，同一层次的包围盒数量大有迥异，包围盒数量越多，图像清晰度越高，但渲染时间复杂度也越高，只有在包围盒数量相当的情况下不同算法性能比较才有意义。比如八叉树的第 6 层和 27 叉树的第 4 层的压缩率对比如表 2 所示。图 13 分别展示了本文算法和文献[28]在 4 个数据集上处理的包围盒数量对比曲线。具体的包围盒数量和体积变换比率如表 3 所示。

对于文献[28]，运用式(5)计算体积变化比率。

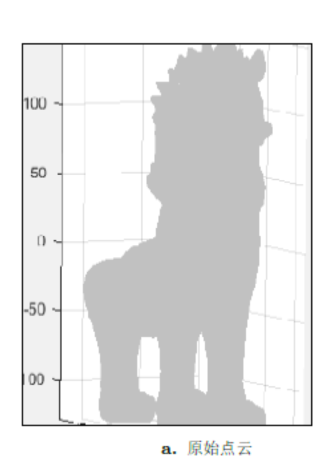
$$\mathrm{OV} = \mathrm{count} * ((1/3)^{depth})^3 \qquad (5)$$

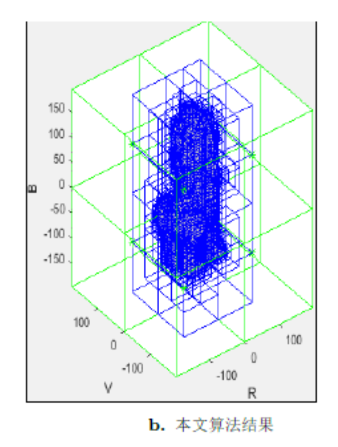
其中：count 是在 depth 层次上的节点/包围盒的数量，$(3^{depth})^3$ 表示完全 27 叉树的节点/包围盒数量。

对于本文算法，应用式(6)计算体积变换比率，$(2^{depth})^3$ 表示完全八叉树的节点/包围盒数量。

$$\mathrm{OV} = \mathrm{count} * \left(\left(\frac{1}{2}\right)^{depth}\right)^3 \qquad (6)$$

图 13 和表 3 中包围盒数量反映了每次分割时绘制立方体的时间成本，可以看出本文算法的优势在于其快速收敛性。体积变换比率表示数据相对于标准完全树网格的扭曲程度。图 14 和表 3 反映了本文算法的数据相对扭曲程度较低，精确度较好。


a. 原始点云


b. 本文算法结果

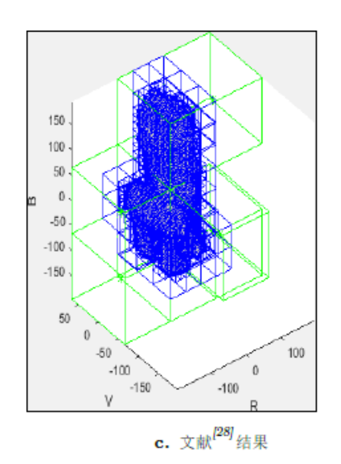

c. 文献[28]结果

图 8 3D 模型 2 种不同分割方案上的分解

表 2　4 个数据集上两种分割方案对应层上的压缩率对比分析

数据集(点数量)	本文算法第 6 层		文献[28]第 4 层	
	节点数量	压缩率	节点数量	压缩率
石狮(309153)	4049	1.306%	6370	2.060%
中国龙(44329)	1745	3.936%	2639	5.953%
太阳花(50826)	2681	5.275%	4281	8.423%
潜艇(360822)	4208	1.166%	6792	1.882%

表 3　4 个数据集上两个分割方案的分解对比分析

(点数量) 数据集 （L_{2max}，x的最大球体半径）			(309153) 石狮 (196.1343)	(44329) 中国龙 (0.1472)	(50826) 太阳花 (9.7326)	(360822) 潜艇 (14.2494)
文献[28]	层次 1	节点数量	8	9	15	10
		体积变化比率	0.2963	0.3333	0.5556	0.3704
	层次 2	节点数量	77	51	68	79
		体积变化比率	0.1056	0.0700	0.0933	0.1084
	层次 3	节点数量	722	390	507	711
		体积变化比率	0.0367	0.0198	0.0258	0.0361
	层次 4	节点数量	6370	2639	4281	6792
		体积变化比率	0.0120	0.0005	0.0081	0.0128
本文算法	层次 1	节点数量	8	8	7	8
		体积变化比率	1. 0	1. 0	0. 8750	1. 0
	层次 2	节点数量	16	16	18	18
		体积变化比率	0. 25	0. 25	0. 2813	0. 2813
	层次 3	节点数量	59	43	51	61
		体积变化比率	0. 1152	0. 0840	0. 0996	0. 1191
	层次 4	节点数量	232	151	185	230
		体积变化比率	0. 0566	0. 0369	0. 0452	0. 0562
	层次 5	节点数量	1012	521	691	1002
		体积变化比率	0. 0309	0. 0159	0. 0211	0. 0306
	层次 6	节点数量	4039	1745	268	4208
		体积变化比率	0. 0154	0. 0067	0. 0102	0. 0161

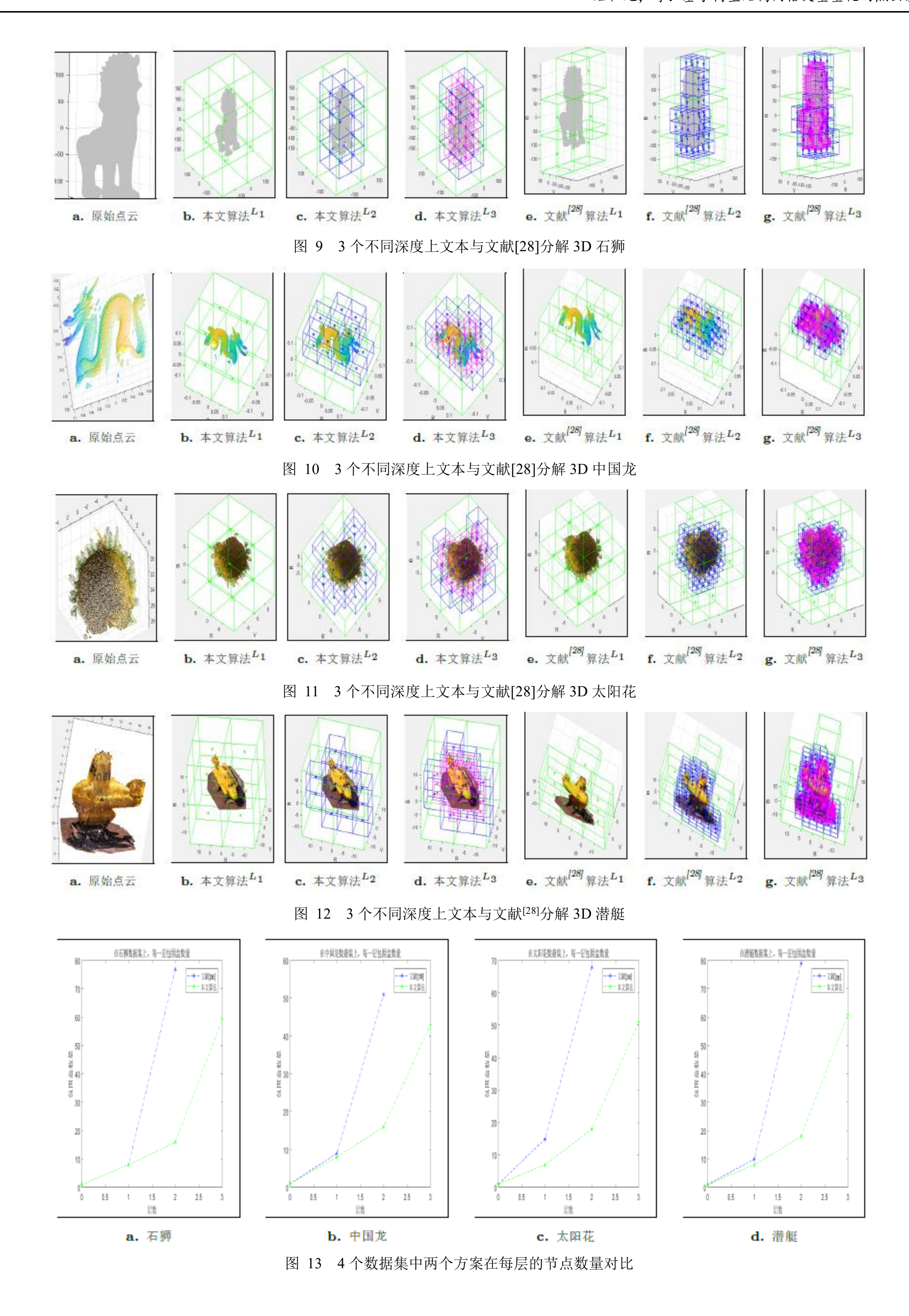


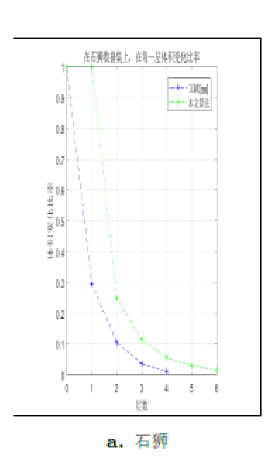
图 9　3 个不同深度上文本与文献[28]分解 3D 石狮

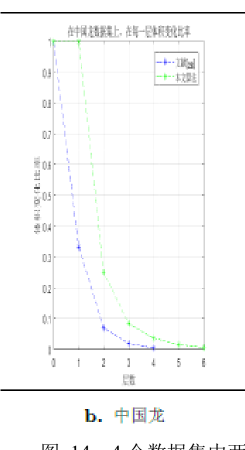
图 10　3 个不同深度上文本与文献[28]分解 3D 中国龙

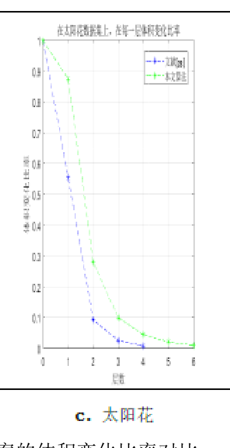
图 11　3 个不同深度上文本与文献[28]分解 3D 太阳花

图 12　3 个不同深度上文本与文献[28]分解 3D 潜艇

图 13　4 个数据集中两个方案在每层的节点数量对比


a. 石狮


b. 中国龙


c. 太阳花

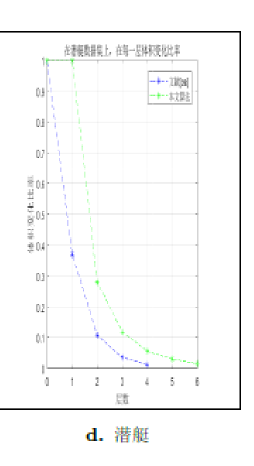

d. 潜艇

图 14　4 个数据集中两个方案的体积变化比率对比

4　结束语

本文提出了一个基于树型结构网格矢量量化的点云渲染算法。算法基于 TSLVQ，用八叉树进行编码，支持在线实时压缩渲染。与现有算法相对比，本文算法在 4 个数据集上均表现出优越性和适用性，能适应不同分辨率，复杂度和精确度需要，具有通用性和可拓展性。这对点云数据的存储、三维重建和增强现实具有明显的工程意义。

未来的工作将集中于动态点云的高效渲染上，以及如何综合时间成本和数据扭曲双重标准，自适应动态选择某种剖分方案以便最佳模拟原始静态点云的形状。

参考文献：

[1] Jensen F V, Nielsen T D. Bayesian networks and decision graphs [M]. New York: Springer, 2007.

[2] Wainwright M J, Jordan M I. Graphical models, exponential families, and variational inference [M]. [S. l.] : Now Publishers Inc. , 2008.

[3] Koller D, Friedman N. Probabilistic graphical models: principles and techniques-adaptive computation and machine learning [M]. Cambridge: MIT Press, 2009.

[4] Edwards D. Introduction to graphical modelling [M]. New York: Springer, 2000.

[5] Bang-Jensen J, Gutin G Z. Digraphs: theory, algorithms and applications [M]. [S. l.] : Springer Publishing Company, 2008.

[6] Kalaiah A, Varshney A. Statistical geometry representation for efficient transmission and rendering [M]. New York: ACM Press, 2005.

[7] Willsky A S. Multiresolution Markov models for signal and image processing [J]. Proceedings of the IEEE, 2002, 90 (8): 1396-1458.

[8] Hornung A, Wurm K M, Bennewitz M, *et al.* OctoMap: an efficient probabilistic 3D mapping framework based on octrees [J]. Autonomous Robots, 2013, 34 (3): 189-206.

[9] 赵尔平，刘炜，党红恩. 海量 3D 点云数据压缩与空间索引技术 [J]. 计算机应用, 2018, 38 (1): 146-151. (Zhao Erping, Liu Wei, Dang Hong'en. Data compression and spatial indexing technology for massive 3D point cloud. Journal of Computer Applications, 2018, 38 (1): 146-151.)

[10] 裴书玉，杜宁，王莉，等. 基于自适应分层的文物点云数据压缩算法 [J]. 计算机应用研究, 2018, 35 (12) . http://www. arocmag. com/article/02-2018-12-062. html. (Pei Shuyu, Du Ning, Wang Li, *et al.* Adaptive slicing-based compression algorithm with cultural relics point clouds. Application Research of Computers, 2018, 35 (12) . [2017-12-08].)

[11] Weir D J, Milroy M J, Bradley C, *et al.* Reverse engineering physical models employing wrap－around B-spline surfaces and quadrics [J]. Journal of Engineering Manufacture, 1996, 210 (22): 147-157.

[12] Martin R R, Stroud I A, Marshall A D. Data reduction for reverse engineering. RECCAD, Deliverable Document 1 COPERNICUS project, No 1068 (1997): 111.

[13] 孟娜. 基于激光扫描点云的数据处理技术研究 [D]. 济南：山东大学, 2009. (Meng Na. Point cloud data process based on laser scanning [D]. Jinan: Shandong University, 2009.

[14] 王平江，黄雪梅. 曲面激光密集测量三维数据的三角片逼近方法 [J]. 图学学报, 1998 (1): 17-27. (Wang Pingjiang, Huang Xuemei. Triangular slice approximation for 3d data with laser intensity measurement [J]. Journal of Graphics, 1988 (1): 17-27.)

[15] Chen Y H, Ng C T, Wang Y Z. Generation of an STL file from 3d measurement data with user-controlled data reduction [J]. International Journal of Advanced Manufacturing Technology, 1999, 15 (2): 127-131.

[16] Taubin G, Zhang T, Golub G. Optimal surface smoothing as filter design [C]// Proc of European Conference on Computer Vision. [S. l.] : Springer-Verlag, 1996: 283-292.

[17] 王向阳，杨红颖，高存臣. 基于小波变换的有损图像压缩算法研究 [J]. 计算机工程与应用, 2001, 37 (15): 82-84. (Wang Xiangyang, Yang Hongying, Gao Cunchen. The image compression based on wavelet transform [J]. Journal of Computer Engineering and Applications, 2001, 37 (15): 82-84.)

[18] Ochotta T, Saupe D. Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields [C]// Proc of Eurographics Conference on Point-based Graphics. [S. l.] : Eurographics Association, 2004: 103-112.

[19] Kammerl J, Blodow N, Rusu R B, *et al.* Real-time compression of point cloud streams [C]// Proc of IEEE International Conference on Robotics and

Automation. 2012: 778-785.

[20] Berger T. Rate-distortion theory [M]. [S. l.] : Prentice-Hall, 1971.

[21] Shannon C E A. A mathematical theory of communication [J]. ACM SIGMOBILE Mobile Computing & Communications Review, 2001, 5 (1): 3-55.

[22] Gersho A, Gray R M. Vector quantization and signal compression [J]. Springer International, 1992, 159 (1): 407-485.

[23] Ricordel V. Etude d'un schéma de quantification vectorielle algébrique et arborescente. Application à la compression de séquences d'images numériques [J]. Ref, 1996.

[24] Conway J H, Sloane N J A. Sphere Packing, Lattices and Groups [J]. American Mathematical Monthly, 1993, 96 (6) .

[25] Linde Y, Buzo A, Gray R M. An Algorithm for Vector Quantizer Design [J]. IEEE Trans on Commun, 1980, 28 (1): 84-95.

[26] Conway J, Sloane N. Fast quantizing and decoding and algorithms for lattice quantizers and codes [J]. IEEE Trans on Inf. Theory, 1982, 28 (2): 227-232.

[27] Ricordel V, Labit C. Vector Quantization by Packing of Embedded Truncated Lattices [M]. OAI, 2010.

[28] Ricordel V, Labit C. Tree-structured lattice vector quantization [C]// Proc of European Signal Processing Conference. 1996: 1-4.

[29] Torralba A, Murphy K P, Freeman T W. The MIT-CSAIL database of objects and scenes [DB]. (2009) . http://people. csail. mit. edu/tmertens/textransfer/data/.

[30] Levoy, Marc, *et al.* The Stanford 3D scanning repository [EB/OL]. (2005) . http://www-graphics. stanford. edu/data/3dscanrep.